

Engineering Optimization as a Problem Solving Tool: Exam I Engineering Optimization ECE 802-604

Saad Bin Qaisar
ECE Department, Michigan State University,
East Lansing, MI- 48824, United States
E-mail: qaisarsa @egr.msu.edu

Abstract –Solved here are three problems for Engineering Optimization Class.

Index Terms – Engineering Optimization, SVM

I. PROBLEM 1

To reformulate the detection problem (for M hypothesis) as an optimization problem, and to show that this re-interpretation also results in the same solution.

A. Hypothesis Testing

$$H_i : y = x_i + n_i, i = 1, 2, \dots, M$$

$$p_i = \text{Noise Distribution}$$

$$M = 2$$

$$p_i(y) = p(y | H_i), i = 1, 2$$

$$\text{Likelihood Ratio } l = \frac{p_2(y)}{p_1(y)}$$

$$\text{Threshold } \tau$$

$$\text{Hypothesis Selection } \begin{cases} H_2 \text{ if } l \geq \tau \\ H_1 \text{ if } l \leq \tau \end{cases}$$

Suppose Y is a random variable with values in $\{1, \dots, n\}$, with a distribution that depends on a parameter $\theta \in \{1, \dots, m\}$. The distributions of Y, for the m possible values of θ , can be represented by a matrix $P \in R^{n \times m}$, with elements:

$$p_{kj} = \text{prob}(Y = k | \theta = j)$$

The jth column of P gives the probability distribution associated with the parameter value $\theta = j$. We consider the problem of estimating θ , based on an observed sample of Y. The m values of θ are called the hypotheses, and finding the best value of θ is hypothesis testing.

A randomized detector of θ is a random variable $\hat{\theta} \in \{1, \dots, m\}$, with a distribution dependent on the observed value of Y. A randomized detector matrix $T \in R^{m \times n}$ with elements

$$t_{ik} = \text{prob}(\hat{\theta} = i | X = k)$$

For a randomized detector defined by the matrix T, we define the detection probability matrix as $D_{m \times m} = TP$. We have:

$$D_{ij} = (TP)_{ij} = \text{prob}(\hat{\theta} = i | \theta = j)$$

The diagonal entries of D, arranged in a vector, give the detection probabilities, and denoted as P^d :

$$P_i^d = D_{ii} = \text{prob}(\hat{\theta} = i | \theta = i)$$

The error probabilities are the complements, and are denoted as P^e :

$$P_i^e = 1 - D_{ii} = \sum_{j \neq i} D_{ji} = \text{prob}(\hat{\theta} \neq i | \theta = i)$$

Thus, the optimal detector design for the multi hypothesis problem can be formulated by introducing a weighting matrix (for scalarization) W for D, where, $W \in R^{m \times m}$ and satisfies:

$$W_{ii} = 0, i = 1, \dots, m,$$

$$W_{ij} > 0, i, j = 1, \dots, m \quad i \neq j$$

W is a weighting matrix, with weight W_{ij} associated with the error of guessing $\hat{\theta} = i$, when in fact $\theta = j$. Thus the multihypothesis detection problem can be rephrased as

$$\text{Minimize } \text{tr}(W^T D)$$

$$\text{Subject to } t_k \geq 0, \mathbf{1}^T t_k = 1, k = 1, \dots, n \quad (1)$$

For Binary Hypothesis Testing

Since m=2 in our problem, it is a case of binary hypothesis testing. Let we are interested in x_2 to occur. Thus, if $y = x_2 + n_2$, we say that our event of interest did occur (positive test). If $y = x_1 + n_1$, we say that event did not occur (negative test). The detection probability matrix $D \in R^{2 \times 2}$ is traditionally expressed as

$$D = \begin{bmatrix} 1 - P_{fp} & P_{fn} \\ P_{fp} & 1 - P_{fn} \end{bmatrix}$$

Where P_{fn} is the probability of false negative (i.e. the test is negative when event has occurred) and P_{fp} is the probability of false positive (i.e. the test is positive when event did not occur).

We assume random variable Y to be generated from one of two distributions, $p \in R^n$ and $q \in R^n$. The optimal trade-off curve between P_{fn} and P_{fp} is called the receiver operating characteristics (ROC), determined by distributions p and q . When $l \leq t$, event did not occur, i.e. $\hat{y} = x_1$. When $l \geq t$, the event did occur, i.e. $\hat{y} = x_2$.

Thus, from our scalarized multi hypothesis testing formulated in (1), we have:

$$\hat{y} = \begin{cases} x_2 & W_{21}p_k \leq W_{12}q_k \\ x_1 & W_{21}p_k \geq W_{12}q_k \end{cases}$$

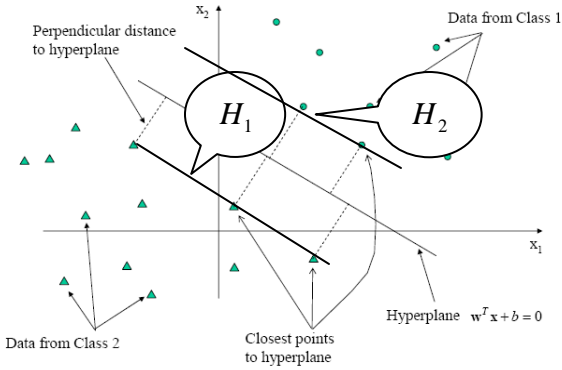
This implies a likelihood ratio test, i.e. ratio $\frac{P_k}{Q_k}$ is more than the

threshold $\frac{W_{12}}{W_{21}}$, the test is negative (i.e. $\hat{y} = x_1$), otherwise, test is

positive. Hence, the reinterpretation also results in the same solution. Similar arguments can be made Bayesian detection, minimax detection, and Neyman Pearson detection.

II. PROBLEM II

Maximizing the separation between the hyperplane and the closest points is equivalent to minimizing $\|w\|$



A. Approach I (Covers 2a,2b,2c,2d)

We have a problem of optimally separating the set of training vectors belonging to two separate classes,

$$D = \{ (x^1, y^1), \dots, (x^l, y^l) \}, \quad x \in R^n, \quad y \in \{1, -1\}$$

with a hyperplane

$$\langle w, x \rangle + b = 0 \quad (1)$$

We intend to maximize the distance between the closest vector and the hyperplane. Without loss of generality, it is appropriate to consider a canonical hyperplane, where the parameters w, b are constrained by:

$$\min |\langle w, x^i \rangle + b| = 1 \quad (2)$$

Thus, it implies that the norm of the weight vector should be equal to the inverse of the distance, of the nearest point in the data set to the hyperplane. A separating hyperplane in canonical form must satisfy following constraints:

$$y^i [\langle w, x^i \rangle + b] \geq 1, \quad i = 1, \dots, l \quad (3)$$

The distance $d(w, b; x)$ of a point x from the hyperplane (w, b) is:

$$d(w, b; x) = \frac{|\langle w, x^i \rangle + b|}{\|w\|} \quad (4)$$

The optimal hyperplane is given by maximizing the margin ρ , subject to constraints of equation (3).

$$\begin{aligned} \rho(w, b) &= \min_{x^i: y^i = -1} d(w, b; x^i) + \min_{x^i: y^i = 1} d(w, b; x^i) \\ &= \min_{x^i: y^i = -1} \frac{|\langle w, x^i \rangle + b|}{\|w\|} + \min_{x^i: y^i = 1} \frac{|\langle w, x^i \rangle + b|}{\|w\|} \\ &= \frac{1}{\|w\|} \left(\min_{x^i: y^i = -1} |\langle w, x^i \rangle + b| + \min_{x^i: y^i = 1} |\langle w, x^i \rangle + b| \right) \\ &= \frac{2}{\|w\|} \end{aligned} \quad (5)$$

Hence the hyperplane that optimally separates the data is the one that minimizes:

$$\Phi(w) = \frac{1}{2} \|w\|^2 \quad (6)$$

The **solution** to the optimization problem of (6) under the constraints of (3) is given using the Lagrange function. Thus,

$$\Phi(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha_i (y^i [\langle w, x^i \rangle + b] - 1) \quad (7)$$

Where α are the Lagrange multipliers. The Lagrangian has to be minimized with respect to w, b , and maximized with respect to $\alpha \geq 0$. Classical Lagrangian duality enables the primal problem, (7), to its dual problem, which is both conceptually and computationally easier to solve. Thus, the dual problem is:

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \left(\min_{w, b} \Phi(w, b, \alpha) \right) \quad (8)$$

The minimum with respect to w and b of the Lagrangian, Φ , is given by,

$$\begin{aligned}\frac{\partial \phi}{\partial b} = 0 &\Rightarrow \sum_{i=1}^l \alpha_i y_i = 0 \\ \frac{\partial \phi}{\partial w} = 0 &\Rightarrow w = \sum_{i=1}^l \alpha_i y_i x_i\end{aligned}\quad (9)$$

Hence, by (7),(8) and (9), the dual problem is given as:

$$\max_{\alpha} (W(\alpha)) = \max_{\alpha} -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{k=1}^l \alpha_k \quad (10)$$

And hence the solution to the problem is given by:

$$\alpha^* = \arg \min_{\alpha} \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_{k=1}^l \alpha_k \quad (11)$$

With constraints,

$$\begin{aligned}\alpha_i &\geq 0 \quad i = 1, \dots, l \\ \sum_{j=1}^l \alpha_j y_j &= 0\end{aligned}\quad (12)$$

Solving equation (11) with constraints equation (12) determines the Lagrange multipliers, and the optimal separating hyperplane is given by:

$$\begin{aligned}w^* &= \sum_{i=1}^l \alpha_i y_i x_i \\ b^* &= -\frac{1}{2} \langle w^*, x_r + x_s \rangle\end{aligned}\quad (13)$$

Where x_r and x_s are any support vectors from each class satisfying,

$$\alpha_r, \alpha_s > 0, \quad y_r = -1, \quad y_s = 1$$

Matlab Solution:

Solving through customized version of [4], and matlab quadratic programming tools, optimal results are:

Results:

Support Vector Solution:

Execution time: 2.2 seconds

Status: OPTIMAL_SOLUTION

$\|w\|^2$: 4.923623

Margin: 0.901338

Sum α : 4.923623

Support Vectors: 2

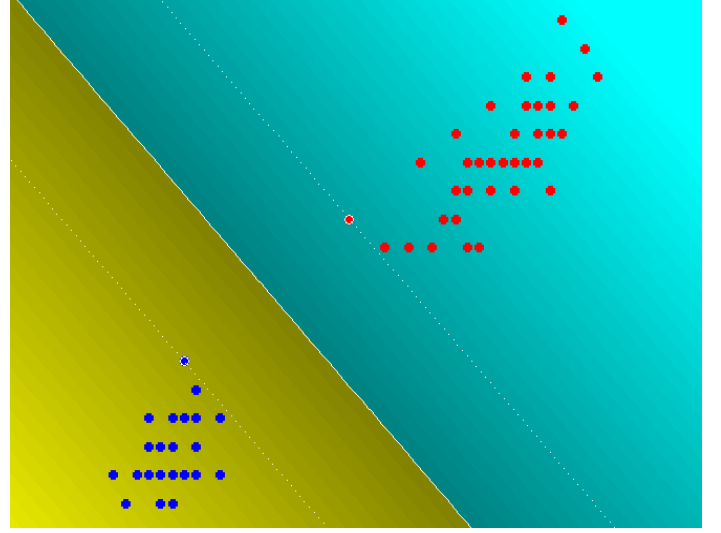


Figure 1: Optimal Hyperplane separating the two data sets

B. Approach II (for 2a,2b)

We assume that hyperplane $H (wx + b = 0)$ is placed at center of the closest points on each side, and the distance of each of the points on both sides from the hyperplane H is equal. Let the closest points be a_1, a_2 with $|a_1| = |a_2| = a$.

Hyperplanes H_1 and H_2 can be thought of passing through a_1, a_2 , respectively, each being parallel to H , and thus, to each other. We formulate an indicator function y_i and thus, have a linear classifier characterized by the set of pairs (w, b) that satisfies the following inequalities for any pattern x_i in the training set.

$$\begin{cases} w^T x_i + b \geq a_2 & \text{if } y_i = +1 \\ w^T x_i + b \leq a_1 & \text{if } y_i = -1 \end{cases}$$

Thus, $y_i (w^T x_i + b) - a \geq 0$

For all points from the hyperplane $H (w^T x + b = 0)$, the distance between origin $(0,0)$ and the hyperplane H is :

$$D = \frac{|w_1 x_1 + w_2 x_2 + b|}{\sqrt{w_1^2 + w_2^2}} = \frac{|b|}{\|w\|_2}$$

The signed distance between Hyperplane $H_1 (w^T x + b \leq a_1)$ and origin is thus $\frac{a_1 - b}{\|w\|_2}$.

The signed distance between Hyperplane $H_2 (w^T x + b \geq a_2)$

and origin is thus $\frac{a_2 - b}{\|w\|_2}$.

Since $|a_1| = |a_2| = a$ (const.), we have separation between

H_1 and H_2 as $\frac{2a}{\|w\|_2}$. From these considerations, it follows that

identification of optimum separation hyperplane is performed by maximizing $\frac{a}{\|w\|_2}$ which is equivalent to minimizing $\frac{\|w\|_2}{a}$,

or, minimizing $\|w\|_2$.

Note: We can reach to similar conclusion by following the approach in Section 8.6.1 Robust Linear Discrimination and Problem 8.23 of [1], with slight modifications of affine functions, and solving the dual problem.

2 (b):

By change of variables $\tilde{w} = w/a$, $\tilde{b} = b/a$, the equivalent convex optimization problem can be written as:

$$\text{Minimize } \|\tilde{w}\|_2$$

$$\text{Subject to } \tilde{w}^T x_i + \tilde{b} \geq 1, \quad y_i = +1$$

$$\tilde{w}^T x_i + \tilde{b} \leq -1, \quad y_i = -1$$

It can also be stated as:

$$\text{Minimize } \|\tilde{w}\|_2$$

$$\text{Subject to } y_i(\tilde{w}^T x_i + \tilde{b}) \geq 1$$

It's a Quadratic Programming problem with affine constraints.

III. PROBLEM III

We are given the set of measurements y that are output of some function $f(x)$.

$$y = f(x) + n$$

Where n corresponds to noise generated according to some probability distribution p . We wish to fit a function \hat{f} to the data

$\{(x_i, y_i)\}_{i=1}^M$ given M input-output pairs. We can formulate multiple minimization criterions some of which include:

Sum Squared Error:

$$\text{Minimize } \sum_{i=1}^M (f(x_i) - \hat{f}(x_i))^2 \quad (1)$$

Mean Squared Error:

$$\text{Minimize } \frac{1}{M} \sum_{i=1}^M (f(x_i) - \hat{f}(x_i))^2 \quad (2)$$

Root Mean Squared Error:

$$\text{Minimize } \sqrt{\frac{1}{M} \sum_{i=1}^M (f(x_i) - \hat{f}(x_i))^2} \quad (3)$$

The problem is not convex in general. The squared difference $(f(x_i) - \hat{f}(x_i))^2$ should be convex over the parameters to be approximated, a, b, c in case of problem 3(b).

Problem 3(b)

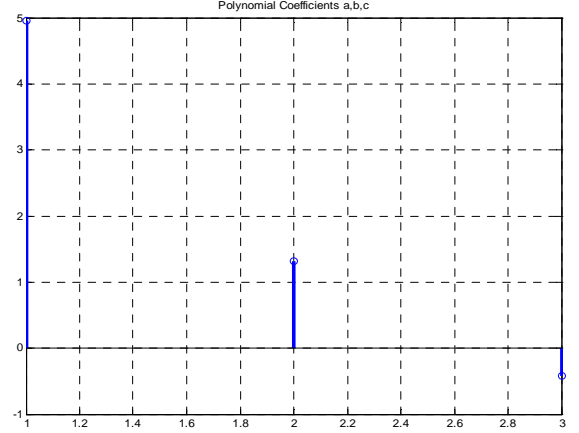


Figure 2: Polynomial Coefficients a,b,c

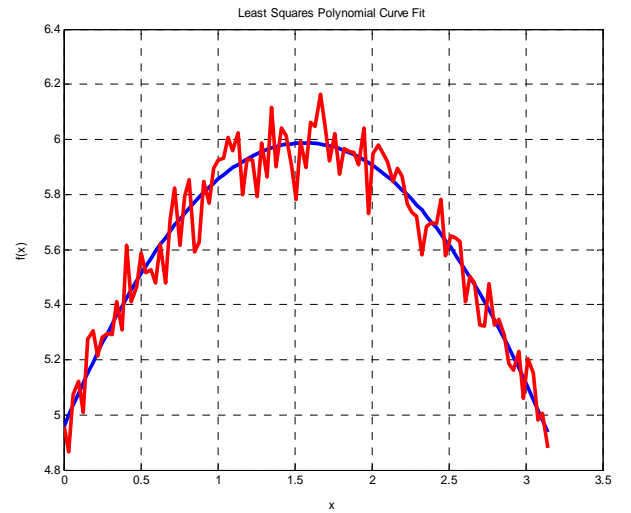


Figure 3: Least Squares Polynomial Fit

Sum Squared Error (SSE) = 0.7313
Mean Squared Error (MSE) = 0.0072
Root Mean Squared Error = 0.0851

See Annex A and attached files for code

REFERENCES

- [1] Stephen Boyd, Convex Optimization, Cambridge University Press, 2006.
- [2] Support Vector Machines, Optimum Separation Hyperplane, http://www.support-vector-machines.org/SVM_osh.html
- [3] Steve R. Gunn, Support Vector Machines for Classification and Regression, TR-1998
- [4] University of Southampton, MATLAB Support Vector Machine Toolbox, <http://www.isis.ecs.soton.ac.uk/resources/svminfo/>

ANNEX A

A. Code for Problem 3

```
% This function calculates the Polynomial Least Squares fitting for the
% data, based upon optimization algorithm:
% http://mathworld.wolfram.com/LeastSquaresFittingPolynomial.html
clear all
load problem3_v6
x=problem3_Data(:,1);
y=problem3_Data(:,2);
n=2;
X(:,n+1) = ones(length(x),1);
% Generating a VanderMonde Matrix
for j = n:-1:1
X(:,j) = x.*X(:,j+1);
end
```

```
X=fliplr(X);
% Getting the Coefficients
a=inv(X'*X)*X'*y;
len_x=length(x);
% Obtaining an Approximation
for(i=1:len_x)
y_new(i)=a(1)+(a(2)*x(i))+(a(3)*x(i)^2);
end
% Plotting the Results
plot(x,y_new);
hold on
plot(x,y,'r')
xlabel('x')
ylabel('f(x)')
title('Least Squares Polynomial Curve Fit');
grid on
hold off
figure;
stem(fliplr(a));
title('Polynomial Coefficients a,b,c')
grid on
y_new=y_new(:);
e=sum((y_new-y).^2); % Error Function
```

B. Code for Problem 2

Please see the attached zipped files.